The support for the amendment to Claims 1, 7, and 18 is found on pages 16-17 and page 10, lines 14-18. The support for the amendment to Claim 13 is found on page 16, lines 5-6 and Figure 11.

Claims 21-44 are newly added. The support for Claims 21, 22, 26, 27, 32, 33, 37, and 38 is found in page 11, lines 15-16. The support for Claims 23, 28, 35, and 39 is found on page 15, lines 22-24. The support for Claims 24, 29, 36, and 40 is found on page 14, line 19. The support for Claims 25, 30, 34, and 41 is found in Figure 4. The support for Claim 31 is found on pages 16-17 and page 10, lines 14-18. The support for Claim 42 is found on page 12, lines 1-14. The support for Claim 43 is found on page 11, lines 17-19. The support for Claim 44 is found on page 10, line 11.

It is respectfully submitted that no new matter has been added.

Specification

The first paragraph has been replaced so that all cited patent applications are referred to by a serial number. It is respectfully submitted that no new matter has been added and respectfully requested that the Patent Office withdraw the objection to the specification.

Claim Rejections - 35 USC § 102

The Patent Office rejected claims 1-20 under 35 USC 102(e) as being anticipated by Hamby, U.S. Patent No..

A claim is anticipated by a reference if each and every element of the claim is taught by the reference or the element is inherent. MPEP 2131.

Claims 1-12, 18-34, and 36-44 recite"the byte-code includes at least one dynamic base object, the at least one dynamic base object comprising an interface dynamic base object and an implementation dynamic base object that communicate with each other

over a message bus." Claims 13-17 and 35 recite " the loader being capable of interpreting, just in time compiling, and pre-compiling."

It is respectfully submitted that these limitations in combination with the other limitations found in the claims are not anticipated nor made obvious by Hamby or the prior art of record.

CONCLUSION

In light of the foregoing, amendments and supporting arguments, reconsideration of all pending claims is requested, and a Notice of Allowance is earnestly solicited.

Respectfully submitted,

GATEWAY, INC.,

By: Walter Malinowski

Walter Malinowski
Reg. No. 43,423

SUITER & ASSOCIATES PC
14301 FNB Parkway, Suite 220
Omaha, Nebraska 68154
Telephone 402.496.0300
Facsimile 402.496.0333

VERSION WITH MARKINGS TO SHOW CHANGES MADE

The present invention claims the benefit under 35 U.S.C. §119(e) of United States Provisional Patent Application Serial Number 60/127,767 filed on April 5, 1999. Said United States Provisional Application 60/127,767 is herein incorporated by reference in its entirety. The present application also claims the benefit under 35 U.S.C. § 120 of United States Patent Application Serial Number 09/312,123, filed May 14, 1999, pending. Said United States Application Serial Number 09/312,123 is herein incorporated by reference in its entirety. The present invention also incorporates the following United States patent applications by reference in their entirety: Serial No. 09/542,559, docket no. P1639US00 (AMI 99-0002), filed April 4, 2000; Serial No. 09/542,716, docket no. P1637US00 (AMI 99-0003), filed April 4, 2000; Serial No. 09/542,743, docket no. P1640US00 (AMI 99-0004), filed April 4, 2000; and Serial No. 09/542,159, docket no. P1641US00 (AMI 99-0005), filed April 4, 2000.

In the claims:

1. (Amended) A method for dynamic compiling, comprising:

loading byte-code on a digital information appliance, said byte-code suitable for including a tagged section;

identifying the tagged section of the byte-code; and

compiling the tagged section of byte-code;

wherein the tagged section is compiled when the byte-code is loaded so as to enable the digital information appliance to utilize the tagged section of byte-code without additional compiling of the tagged section of byte-code by the digital information appliance.

wherein the byte-code includes at least one dynamic base object, the at least one dynamic base object comprising an interface dynamic base object and an implementation dynamic base object that communicate with each other over a message bus.

7. (Amended) A digital information appliance suitable for dynamic coupling, comprising:

a processor for implementing a program of instructions; and

a memory for storing the program of instructions, the program of instructions suitable for configuring the digital information appliance to load byte-code, said byte-code suitable for including a tagged section;

identify the tagged section of the byte-code; and

compile the tagged section of byte-code;

wherein the tagged section is compiled when the byte-code is loaded so as to enable the digital information appliance to utilize the tagged section of byte-code without additional compiling of the tagged section of byte-code by the digital information appliance,

wherein the byte-code includes at least one dynamic base object, the at least one dynamic base object comprising an interface dynamic base object and an implementation dynamic base object that communicate with each other over a message bus.

13. (Amended) A system for providing an execution environment that is suitable for dynamic compiling, comprising:

a memory device suitable for storing computer readable information;

a loader coupled to the memory device, the loader suitable for loading byte-code to the memory, said byte-code suitable for including a tagged section, the loader being capable of interpreting, just in time compiling, and pre-compiling;

an identifier coupled to the loader, the identifier suitable for identifying the tagged section of the byte-code;

a compiler coupled to the identifier;

wherein the identified tagged section is compiled by the compiler when the byte-code is loaded so as to enable the tagged section of byte-code to be utilized without additional compiling of the tagged section of byte-code.

18. (Amended) A method for providing an execution environment in an information appliance network, comprising:

a) encoding an application source code in a processor independent byte-code;

b) tagging at least some portion of said processor independent byte-code; and

c) compiling at least some portion of said tagged processor independent byte-code,

wherein the independent byte-code includes at least one dynamic base object, the at least one dynamic base object comprising an interface dynamic base object and an implementation dynamic base object that communicate with each other over a message bus.